Optimized Framework based on Rough Set Theory for Big Data Pre-processing in Certain and Imprecise Contexts" -- Marie Sklodowska-Curie Project: Open Problems'

Chelly Dagdia, Zaineb

DOI:
10.4230/DagRep.7.9.62

Publication date:
2018

Citation for published version (APA):
Chelly Dagdia, Z. (2018). Optimized Framework based on Rough Set Theory for Big Data Pre-processing in Certain and Imprecise Contexts" -- Marie Sklodowska-Curie Project: Open Problems'. Abstract from Recent Trends in Knowledge Compilation, Dagstuhl, Germany. https://doi.org/10.4230/DagRep.7.9.62

Report from Dagstuhl Seminar 17381

Recent Trends in Knowledge Compilation

Edited by

Adnan Darwiche[1], Pierre Marquis[2], Dan Suciu[3], and Stefan Szeider[4]

1 **UCLA, US,** darwiche@cs.ucla.edu
2 **Artois University – Lens, FR,** marquis@cril.univ-artois.fr
3 **University of Washington – Seattle, US,** suciu@cs.washington.edu
4 **TU Wien, AT,** sz@ac.tuwien.ac.at

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 17381 "Recent Trends in Knowledge Compilation".

Seminar September 17–22, 2017 – http://www.dagstuhl.de/17381
1998 ACM Subject Classification Artificial Intelligence / Robotics, Data Bases / Information Retrieval, Data Structures / Algorithms / Complexity
Keywords and phrases Knowledge compilation, Constraints, Preprocessing, Probabilistic databases, Model counting
Digital Object Identifier 10.4230/DagRep.7.9.62
Edited in cooperation with Friedrich Slivovsky

1 Executive summary

Adnan Darwiche
Pierre Marquis
Dan Suciu
Stefan Szeider

Knowledge compilation (KC) is a research topic which aims to investigate the possibility of circumventing the computational intractability of hard tasks, by preprocessing part of the available information, common to a number of instances. Pioneered almost three decades ago, KC is nowadays a very active research field, transversal to several areas within computer science. Among others, KC intersects knowledge representation, constraint satisfaction, algorithms, complexity theory, machine learning, and databases.

The results obtained so far take various forms, from theory (compilability settings, definition of target languages for KC, complexity results, succinctness results, etc.) to more practical results (development and evaluation of compilers and other preprocessors, applications to diagnosis, planning, automatic configuration, etc.). Recently, KC has been positioned as providing a systematic method for solving problems beyond NP, and also found applications in machine learning.

The goal of this Dagstuhl Seminar was to advance both aspects of KC, and to pave the way for a fruitful cross-fertilization between the topics, from theory to practice.

The program included a mixture of long and short presentations, with discussions. Several long talks with a tutorial flavor introduced the participants to the variety of aspects in knowledge compilation and the diversity of techniques used. System presentations as well as an open problem session were also included in the program.

2 Table of Contents

3 Talks

3.1 A Circuit-Based Approach to Efficient Enumeration

Antoine Amarilli (Télécom ParisTech, FR)

Joint work of Antoine Amarilli, Pierre Bourhis, Louis Jachiet, Stefan Mengel
Main reference Antoine Amarilli, Pierre Bourhis, Louis Jachiet, Stefan Mengel: "A Circuit-Based Approach to Efficient Enumeration", in Proc. of the 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland, LIPIcs, Vol. 80, pp. 111:1–111:15, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
URL http://dx.doi.org/10.4230/LIPIcs.ICALP.2017.111

Enumeration is the task of computing efficiently multiple solutions to a computational problem, one after another. This task has been studied in knowledge compilation, where we compile problems to circuits and enumerate their satisfying assignments. However, it has also been studied in database theory and logics to show constant-delay enumeration results for query evaluation, e.g., for monadic second-order logic on trees.

We show how to recapture these database theory results via a knowledge compilation approach. To do so, we compile the database query to a circuit that represents all its answers in a factorized way. We then present an efficient algorithm to enumerate the satisfying valuations of the circuit. Our algorithm works on structured d-DNNFs and focuses on valuations of constant Hamming weight to achieve the constant-delay bound.

3.2 Knowledge Compilation: Representations and Lower Bounds

Paul Beame (University of Washington – Seattle, US)

Joint work of Jerry Li, Vincent Liew, Sudeepa Roy, Dan Suciu

We survey a variety of representations of Boolean functions and summarize the connections between these representations and exact model counting algorithms.

We then discuss our work showing a variety of lower bounds for several of these representations, including decision-DNNFs, SDDs, and DNNFs, as well as applications of those lower bounds.

Finally, we discuss a more general representation suitable for exact model counting for which we are not aware of any lower bounds for explicit Boolean functions.

3.3 Provenance for tree-like instances: a bridge between database and knowledge compilation

Pierre Bourhis (CNRS – Lille, FR)

The notion of provenance is well known notion in database to explain the contribution of the data to the computation of the query. This notion has some practical use in particular in probabilistic data evaluation. In this context, Jha and Suciu showed that if the provenance

can be expressed by well-known classes of circuits then probabilistic data evaluation becomes tractable. Following this path, we show that the provenance of expressive queries over trees can be expressed by a well-known class of circuits d-DNNF and then derive a set of interesting properties from this result.

3.4 Circuit Treewidth, Sentential Decision, and Query Compilation

Simone Bova (TU Wien, AT)

Joint work of Simone Bova, Stefan Szeider
Main reference Simone Bova, Stefan Szeider: "Circuit Treewidth, Sentential Decision, and Query Compilation", in Proc. of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017, pp. 233–246, ACM, 2017.
URL http://dx.doi.org/10.1145/3034786.3034787

The evaluation of a query over a probabilistic database boils down to computing the probability of a suitable Boolean function, the lineage of the query over the database. The method of query compilation approaches the task in two stages: first, the query lineage is implemented (compiled) in a circuit form where probability computation is tractable; and second, the desired probability is computed over the compiled circuit. A basic theoretical quest in query compilation is that of identifying pertinent classes of queries whose lineages admit compact representations over increasingly succinct, tractable circuit classes. In this talk, we focus on queries whose lineages admit circuit implementations with small treewidth, and investigate their compilability within tame classes of decision diagrams. In perfect analogy with the characterization of bounded circuit pathwidth by bounded OBDD width (Jha and Suciu, 2012), we show that a class of Boolean functions has bounded circuit treewidth if and only if it has bounded SDD width. Sentential decision diagrams (SDDs) are central in knowledge compilation, being essentially as tractable as OBDDs but exponentially more succinct. By incorporating constant width SDDs and polynomial size SDDs, we refine the panorama of query compilation for unions of conjunctive queries with and without inequalities.

3.5 Updating the Knowledge Compilation Map

Simone Bova (TU Wien, AT)

Joint work of Simone Bova, Florent Capelli, Stefan Mengel, Friedrich Slivovsky
Main reference Simone Bova, Florent Capelli, Stefan Mengel, Friedrich Slivovsky: "Knowledge Compilation Meets Communication Complexity", in Proc. of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, pp. 1008–1014, IJCAI/AAAI Press, 2016.
URL http://www.ijcai.org/Abstract/16/147

Introduced by Darwiche and Marquis (2002), the knowledge compilation map offers a systematic framework to assess alternative representations of propositional theories relative to their succinctness and tractability. In this tutorial talk we exploit the recently established connection between knowledge compilation and communication complexity to update and refine the information originally incorporated in the knowledge compilation map.

3.6 Non FPT Lower Bounds for Restricted Decision DNNF

Florent Capelli (INRIA Lille, FR)

Many algorithms are known to exploit the structure of a CNF formula to solve #SAT efficiently on some families of formulas. In particular, for every k, we know how to solve #SAT in time $2^{\Omega(k)} \cdot poly(|F|)$ for a CNF F whose incidence graph has treewidth k. However, the algorithm used to solve #SAT in this case is very different from the one used by most practical tools solving #SAT, which are based on #DPLL, a generalisation of DPLL. It is known that such tools can efficiently exploit the fact that the treewidth of the primal graph of F is bounded but it is not clear if such tools can exploit the incidence treewidth of the formula.

In this talk, we give a partial answer to this question by showing that some natural restrictions of #DPLL cannot solve formulas whose incidence graph has treewidth k.

3.7 Knowledge compilation and compression using interval representations of Boolean functions.

Ondrej Cepek (Charles University – Prague, CZ)

Joint work of Radek Husek, Ondrej Cepek
Main reference Ondrej Cepek, Radek Husek: "Recognition of tractable DNFs representable by a constant number of intervals", Discrete Optimization, Vol. 23, pp. 1–19, 2017.
URL http://dx.doi.org/10.1016/j.disopt.2016.11.002

In this talk we shall focus on a less common way how to represent Boolean functions, namely on representations by intervals of truepoints or by closely related representation by switch-lists. Let f be a Boolean function and let us fix some order of its n variables. The input binary vectors can be now thought of as binary numbers (with bits in the prescribed order) ranging form 0 to $2^n - 1$. An interval representation is then an abbreviated TT (truth table) or MODS (models) representation, where instead of writing out all the input vectors (binary numbers) with their function values, we write out only those binary numbers x for which $f(x) = 1$ (x is a truepoint of f) and simultaneously $f(x-1) = 0$ ($x-1$ is a falsepoint of f) and those binary numbers y for which $f(y) = 1$ (y is a truepoint of f) and simultaneously $f(y+1) = 0$ ($y+1$ is a falsepoint of f). Thus the function is represented by an ordered list of such pairs $[x, y]$ of integers, each pair specifying one interval of truepoints. Note that $x = y$ for those pairs which represent an interval with a single truepoint. In this talk we shall also consider a related representation by switch-lists. A switch is a vector (binary number) x such that $f(x-1) \neq f(x)$. A switch-list is an ordered list of all switches of a given function (under a given order of variables).

There are two natural problems connected to such representation: (1) a knowledge compilation problem, i.e. a problem of transforming a given representation of a Boolean function (Boolean formula, binary decision diagram, Boolean circuit ...) into an interval or switch-list representation, and (2) a knowledge compression problem, i.e. a problem of finding the shortest interval or switch-list representation among those which represent the given function (which amounts to finding such an order of variables). We will summarize

known results about these two problems and present generalizations in both areas. The main result is a polynomial time algorithm that for a Boolean function (given by a tractable formula) outputs a shortest interval and switch-list representations, provided that the number of switches (intervals) is bounded by a constant. This algorithm can be also thought of as a polynomial time recognition algorithm for the class of k-switch (or k-interval) functions given by a tractable formula for any fixed k.

3.8 Optimized Framework based on Rough Set Theory for Big Data Pre-processing in Certain and Imprecise Contexts

Zaineb Chelly Dagdia (Aberystwyth University, GB)

Joint work of Christine Zarges, Gaël Beck, Mustapha Lebbah, Juan J. Merelo Guervós

Over the last decades, the amount of data has increased in an unprecedented rate, leading to a new terminology: "Big Data". Big data are specified by their Volume, Variety, Velocity and by their Veracity/Imprecision. Based on these 4V specificities, it has become difficult to quickly acquire the most useful information from the huge amount of data at hand. Thus, it is necessary to perform data pre-processing as a first step. In spite of the existence of many techniques for this task, most of the state-of-the-art methods require additional information for thresholding and are neither able to deal with the big data veracity aspect nor with their computational requirements. We aim at presenting current progress and insights of this Marie Skłodowska Curie project that aims at filling the mentioned major research gaps by proposing solutions based on Rough Set Theory for data pre-processing and Randomized Search Heuristics for optimization. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 702527.

3.9 Using Knowledge Compilation to Solve PP^{PP}-Complete Problems

YooJung Choi (UCLA, US)

Knowledge compilation has been successfully used to solve "Beyond NP" problems, including some PP-complete and NP^{PP}-complete probabilistic queries. This approach compiles problems into Boolean circuits with certain properties that guarantee linear-time query evaluation. This talk will discuss how knowledge compilation can be used to answer queries that are complete for the more intractable complexity class PP^{PP}. In particular, we show how to solve a canonical PP^{PP}-complete problem MAJ-MAJ-SAT in linear time by compiling the problem instance into a special type of Sentential Decision Diagrams (SDDs) called a constrained SDD. In addition, to demonstrate the practical value of the proposed approach, we will adapt it to answer the Same-Decision Probability (SDP) and Expected Same-Decision Probability (E-SDP) queries for Bayesian networks, which quantify the robustness of threshold-based decisions and have been proposed as stopping and selection criteria for Bayesian decision making.

3.10 A Framework for Parameterized Compilability

Ronald de Haan (University of Amsterdam, NL)

Joint work of Ronald de Haan, Simone Bova, Neha Lodha, Stefan Szeider
Main reference Simone Bova, Ronald de Haan, Neha Lodha, Stefan Szeider: "Positive and Negative Results for Parameterized Compilability," Technical report AC-TR-16-003, Algorithms and Complexity Group, TU Wien, 2016.
URL https://www.ac.tuwien.ac.at/files/tr/ac-tr-16-003.pdf

Many fundamental problems do not admit a polynomial-size compilation that allows queries to be answered in polynomial time. A typical example of such a problem is the problem of deciding whether a given CNF formula (offline input) entails a given clause (online query). The compilability framework of Cadoli, Donini, Liberatore and Schaerf [1] allows us to investigate when poly-size compilations are possible, and when not.

The perspective of parameterized complexity makes it possible to extend the notion of feasible compilability. In addition to the input size, we consider a problem parameter. We then aim to compile the offline instance into an fpt-size compilation, such that online queries can be answered in fpt-time. This could improve the potential of the approach of compilation.

To investigate the boundary between what is fpt-size compilable and what is not, a different theoretical framework is needed. Chen initiated the development of such a framework [2], that we recently extended. In this talk, we explain this framework, and how it can be used. As a running example, we use the problem of clause entailment for CNF formulas. We present both positive and negative parameterized compilability results for various parameters for clause entailment.

References

1 Cadoli, M.; Donini, F. M.; Liberatore, P.; and Schaerf, M. 2002. Preprocessing of intractable problems. *Information and Computation* 176(2):89–120.
2 Chen, H. 2005. Parameterized compilability. In *Proceedings of IJCAI 2005, the 19th International Joint Conference on Artificial Intelligence.*

3.11 A Structural Approach to Model Counting

Robert Ganian (TU Wien, AT)

Joint work of Robert Ganian, Stefan Szeider

The boolean satisfiability problem (SAT) and its generalization to model counting (#SAT) are fundamental problems with a strong overlap into knowledge compilation. One of the basic approaches used to design exact algorithms for these problems is to exploit the structure of variable-clause interactions captured in terms of various graph representations of CNF formulas. The most successful way of doing this is with the use of the structural parameter called treewidth. After reviewing the fundamentals of how treewidth can be used to solve SAT and #SAT, we proceed to provide an overview of very recent advances made in this direction.

References

1 Robert Ganian and Stefan Szeider. *New Width Parameters for Model Counting.* SAT 2017

3.12 Colour Refinement: A Simple Partitioning Algorithm with Applications From Graph Isomorphism Testing to Machine Learning

Martin Grohe (RWTH Aachen, DE)

Colour refinement is a simple algorithm that partitions the vertices of a graph according their "iterated degree sequences". It has very efficient implementations, running in quasilinear time, and a surprisingly wide range of applications. The algorithm has been designed in the context of graph isomorphism testing, and it is used an important subroutine in almost all practical graph isomorphism tools. Somewhat surprisingly, other applications in machine learning, probabilistic inference, and linear programming have surfaced recently.

In the first part of my talk, I will introduce the basic algorithm as well as higher dimensional extensions known as the k-dimensional Weisfeiler-Lehman algorithm. I will also discuss an unexpected connection between colour refinement and a natural linear programming approach to graph isomorphism testing. In the second part of my talk, I will discuss various applications of colour refinement.

3.13 Efficiently Enumerating Minimal Triangulations

Batya Kenig (Technion – Haifa, IL)

Joint work of Nofar Carmeli, Benny Kimelfeld
Main reference Nofar Carmeli, Batya Kenig, Benny Kimelfeld: "Efficiently Enumerating Minimal Triangulations", in Proc. of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017, pp. 273–287, ACM, 2017.
URL http://dx.doi.org/10.1145/3034786.3056109

We present an algorithm that enumerates all the minimal triangulations of a graph in incremental polynomial time. Consequently, we get an algorithm for enumerating all the proper tree decompositions, in incremental polynomial time, where "proper" means that the tree decomposition cannot be improved by removing or splitting a bag. The algorithm can incorporate any method for (ordinary, single result) triangulation or tree decomposition, and can serve as an anytime algorithm to improve such a method. We describe an extensive experimental study of an implementation on real data from different fields. Our experiments show that the algorithm improves upon central quality measures over the underlying tree decompositions, and is able to produce a large number of high-quality decompositions.

3.14 On Compiling (Online) Combinatorial Learning Problems

Frederic Koriche (Artois University – Lens, FR)

In various machine learning applications, the learning problem can be viewed as an interactive process, or repeated game, between the learner and its environment. During each trial of

the game, the learner chooses a predictor from its hypothesis class, and simulaneously, the environment selects an instance. Then, the instance is presented to the learner, which incurs a loss. In this online learning framework, everything is fine when all components of the game are convex. Yet, for many learning tasks of interest in AI, the hypothesis class is combinatorial by nature, and hence, convex optimization algorithms cannot be applied to this setting.

This talk will address the following question: what about compiling the hypothesis space?

Knowledge compilation indeed looks promising for investigating (online) combinatorial learning problems, by compiling a probability distribution on the hypothesis class into an appropriate structure, and by iteratively updating this structure according to incoming instances. We will illustrate this approach by considering several online combinatorial learning problems, including learning conjunctive functions, and graphical probabilistic models.

3.15 Solving very hard SAT problems: a form of partial KC

Oliver Kullmann (Swansea University, GB)

We (Marijn Heule, Victor Marek and me) solved in 2016 the Boolean Pythagorean Triples Problem https://en.wikipedia.org/wiki/Boolean_Pythagorean_triples_problem via SAT. In the first half of the talk I explain the problem, in the second part I explain the underlying method for such very hard (but relatively small) problems, the "Cube-and-Conquer" (C&C) paradigm. The outstanding feature of C&C is that it combines “old and new” (DPLL and CDCL), and I will discuss the fundamental intuitions for this symbiosis.

3.16 SAT by MaxSAT: From NP to Beyond NP and Back Again

Joao Marques-Silva (University of Lisbon, PT)

This talks starts by briefly overviewing recent work on practically efficient algorithms for maximum satisfiability, namely algorithms that iteratively compute unsatisfiable cores and, among these, those that exploit minimum hitting sets. Afterwards, the talk investigates reductions of decision problems to Horn Maximum Satisfiability. It is shown that such reductions enable polynomial refutations of the pigeonhole principle. This in turn motivates a number of research questions.

3.17 Probabilistic Logic Programming and Knowledge Compilation

Wannes Meert (KU Leuven, BE)

Joint work of Jonas Vlasselaer, Guy Van den Broeck, Anton Dries, Angelika Kimmig, Hendrik Blockeel, Jesse Davis, Luc De Raedt

In this talk I'll present the ProbLog probabilistic logic programming system. First, I will introduce the semantics, methodology and system. Thus wat is available today. Second, I will focus on some of our current progress and challenges. This includes incremental compilation (e.g. approximate inference), continuous observations (e.g. sensor measurements), and compiling resource-aware circuits (e.g. memory or energy).

https://dtai.cs.kuleuven.be/problog

3.18 A Dichotomy for Compiling Constraint Satisfaction Problems to Read-Once Decision Diagrams

Stefan Mengel (CNRS, CRIL – Lens FR)

In this talk I presented ongoing work on a dichotomy for compiling constraint satisfaction problems into read-once decision programs that states the following: For every finite constraint language Γ one of the following is true:

- the satisfying assignments of any Γ-formula can be compiled into a multivalued, ordered decision diagram (MODD) in polynomial time, or
- there are Γ-formulas formulas such that compiling them into MODD takes exponential size.

I give the criterion that determines which of the two statements is true. Moreover, I instantiate the dichotomy for the Boolean domain and discuss possible extensions.

This is unpublished, ongoing work.

3.19 BDD/ZDD-based knowledge indexing and real-life applications

Shin-ichi Minato (Hokkaido University, JP)

URL http://art.ist.hokudai.ac.jp/~minato/

BDD (Binary Decision Diagram) [1] is a classical data structure for representing a Boolean function. BDD-based algorithms were developed mainly for VLSI logic design in early 1990s. ZDD (Zero-suppressed BDD) [2] is a variant of BDD, customized for representing a set of combinations, often appear in solving combinatorial problems. BDDs and ZDDs have become more widely known since D. Knuth intensively discussed them in his famous series of books in 2009 [3]. Although a quarter of a century has passed since the original idea of BDD-based operations by R. Bryant [1], there are still many interesting research topics ongoing. One of the most important topics is a fast algorithm of constructing a ZDD which

compactly indexes all the paths in a given graph structure, presented by Knuth [3]. This work is important because many kinds of practical problems are efficiently solved by some variations of this algorithm. In this talk, I will give an overview of the brief history and the basic techniques on BDDs/ZDDs. We then look over some recent research topics from the view point of knowledge indexing.

References

1 R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Transactions on Computers*, vol. C-35, no. 8, pp. 677–691, 1986.
2 S. Minato, "Zero-suppressed BDDs for set manipulation in combinatorial problems," in *Proc. of 30th ACM/IEEE Design Automation Conference (DAC'93)*, 1993, pp. 272–277.
3 D. E. Knuth, *The Art of Computer Programming: Bitwise Tricks & Techniques; Binary Decision Diagrams*. Addison-Wesley, 2009, vol. 4, fascicle 1.

3.20 Efficient Representations for the Modal Logic S5

Alexandre Niveau (Caen University, FR)

Joint work of Niveau, Alexandre; Zanuttini, Bruno
Main reference Alexandre Niveau, Bruno Zanuttini: Efficient Representations for the Modal Logic S5. IJCAI 2016
URL http://www.ijcai.org/Abstract/16/177

In this talk, I presented our ongoing investigation about efficient representations of subjective formulas in the modal logic of knowledge, $\mathsf{S5}$, and more generally of sets of sets of propositional assignments. One motivation for this work is contingent planning, for which many approaches use operations on such formulas, and can clearly take advantage of efficient representations.

This study was started by Bienvenu, Fargier & Marquis [1], who introduced the parameterized language $\mathsf{S5} - \mathsf{DNF}_{\mathsf{L},\mathsf{L}'}$ (L and L' being the languages used at the propositional level). In a paper last year [2], we introduced the language of Epistemic Splitting Diagrams, with the aim of providing more compact representations, and we compared it to two promising instances of $\mathsf{S5} - \mathsf{DNF}_{\mathsf{L},\mathsf{L}'}$, both from the complexity-theoretic viewpoint of the knowledge compilation map and also through preliminary experiments inspired by contingent planning.

References

1 Meghyn Bienvenu, Hélène Fargier, & Pierre Marquis (2010). "Knowledge Compilation in the Modal Logic S5." In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010).
2 Alexandre Niveau & Bruno Zanuttini (2016). "Efficient Representations for the Modal Logic S5." In: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016).

3.21 In-Database Factorized Learning

Dan Olteanu (University of Oxford, GB)

Main reference Mahmoud Abo Khamis, Hung Q. Ngo, XuanLong Nguyen, Dan Olteanu, Maximilian Schleich: "In-Database Learning with Sparse Tensors", CoRR, Vol. abs/1703.04780, 2017.
URL http://arxiv.org/abs/1703.04780

In this talk I will overview work on compilation of join queries into lossless factorized representations. The primary motivation for this compilation is to avoid redundancy in the representation of query results in relational databases. The relationship between the standard listing representation of relations and equivalent factorized representations is on a par with the relationship between a propositional formula in disjunctive normal form and equivalent circuits. For any join query, we give asymptotically tight bounds on the size of its factorized results and on the time to compute them. We show that these factorized results can be arbitrarily more succinct than their equivalent listing representations.

I will also discuss an application of factorized joins to learning regression models over databases. On-going joint work [1] with LogicBlox collaborators shows that in-database factorized learning can speed up real analytical workloads in the retailer domain by several orders of magnitude over state-of-the-art systems.

This work is based on long-standing collaboration with Maximilian Schleich (Oxford) and Jakub Zavodny (Oxford) and more recent collaboration with Mahmoud Abo Khamis (LogicBlox), Hung Ngo (LogicBlox), and XuanLong Nguyen (Michigan).

References

1 Mahmoud Abo Khamis, Hung Q. Ngo, XuanLong Nguyen, Dan Olteanu, and Maximilian Schleich. In-database learning with sparse tensors. *CoRR*, abs/1703.04780, 2017. To appear in ACM PODS 2018.

3.22 Partial matching width and lower bounds for read-once branching programs

Igor Razgon (Birkbeck, University of London, GB)

Many algorithms are known to exploit the structure of a CNF formula to solve #SAT efficiently on some families of formulas. In particular, for every k, we know how to solve #SAT in time $2^{\Omega(k)} \cdot poly(|F|)$ for a CNF F whose incidence graph has treewidth k. However, the algorithm used to solve #SAT in this case is very different from the one used by most practical tools solving #SAT, which are based on #DPLL, a generalisation of DPLL. It is known that such tools can efficiently exploit the fact that the treewidth of the primal graph of F is bounded but it is not clear if such tools can exploit the incidence treewidth of the formula.

In this talk, we give a partial answer to this question by showing that some natural restrictions of #DPLL cannot solve formulas whose incidence graph has treewidth k.

3.23 From knowledge of humans performing everyday activities to the service robots

Lisset Y. Salinas Pinacho (ifib – Bremen, DE)

Joint work of Alexander Wich, Andrei Haidu, Michael Beetz

Humans perform daily manipulation tasks as a routine, which might not present a hight level of complexity while performing, because they poses commonsense knowledge that was acquired since infancy. However, a system which does not own the enormous amount of information that humans have, when trying to perform the same tasks, faces a huge challenge. This kind of system have been named service robots, which are expected to perform autonomously and accurately to be accepted as human's companions. Even when this robots present high manipulation performance nowadays, they are still unable to perform as well as humans. Part of the challenge while performing a daily task successfully is to find the right sources of the required knowledge. Some available sources of commonsense knowledge are, which are unfortunately still incomplete, knowledge bases and datasets from either humans or robots performing a task. By following this idea, the commonsense knowledge obtained in this work comes from different sources as the KnowRob knowledge base [1] and knowledge of humans performing daily tasks, extracted from videos and virtual reality environments [2]. The knowledge from these last mentioned sources has still to be represented and available for later use of service robot. In this work is presented the representation structure using the ideas presented by the KnowRob system, and the knowledge retrieve system. As well as the open platform openEASE [3] to analyse and visualize the data is presented.

References

1 M. Tenorth and M. Beetz, "Knowrob – knowledge processing for autonomous personal robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4261–4266, October 2009.
2 A. Haidu and M. Beetz, "Action recognition and interpretation from virtual demonstrations," in *International Conference on Intelligent Robots and Systems (IROS)*, (Daejeon, South Korea), pp. 2833–2838, 2016.
3 M. Beetz, M. Tenorth, and J. Winkler, "Opean-ease: A knowledge processing service for robots and robotics/ ai researchers," tech. rep., TZI-Bericht 74, Universität Bremen, 2014.

3.24 Compiling Deep Nets

Scott Sanner (University of Toronto, CA)

Aside from standard uses of deep neural networks for supervised and unsupervised learning, we may want to reason about the learned deep network itself. For example, we may wish to know what completion of a partial input maximizes the (probability of an) output. Or if we train deep a generative model, we may wish to make arbitrary probabilistic queries with arbitrary evidence on this model without resorting to sampling. But how do we compile deep networks to representations amenable to efficient MPE, MAP, or marginal query style inferences? In this talk, I will present some initial directions for such deep network compilations.

3.25 Provenance Circuits and Knowledge Compilation in ProvSQL

Pierre Senellart (ENS – Paris, FR)

This talk describes the ProvSQL system, whose goal is to add support for (m-)semiring provenance and uncertainty management to PostgreSQL databases, in the form of a PostgreSQL extension/module/plugin. It is work in progress at the moment.

Probability computation in ProvSQL relies knowledge compilation tools, such as c2d, d4, or dsharp, to compile a circuit expressing the provenance of a query into a d-DNNF, whose probability can be evaluated. We explain this process in this talk and demonstrate the specific use case of knowledge compilation for probabilistic data management.

ProvSQL is freely available for download from https://github.com/PierreSenellart/provsql

3.26 Just-In-Time Compilation of Knowledge Bases

Laurent Simon (University of Bordeaux, FR)

Joint work of Laurent Simon, Gilles Audemard, Jean-Marie Lagniez

Since the first principles of Knowledge Compilation (KC), most of the work has been focused in finding a good compilation target language in terms of compromises between compactness and expressiveness. The central idea remained unchanged in the last fifteen years: an off-line, very hard, stage, allows to "compile" the initial theory in order to guarantee (theoretically) an efficient on-line stage, on a set of predefined queries and operations. We propose a new "Just-in-Time" approach for KC. Here, any knowledge Base (KB) will be immediately available for queries, and the effort spent on past queries will be partly amortized for future ones. To guarantee efficient answers, we rely on the tremendous progresses made in the practical solving of SAT and incremental SAT applicative problems. Even if each query may be theoretically hard, we show that our approach outperforms previous KC approaches on the set of classical problems used in the field, and allows to handle problems that are out of the scope of current approaches.

3.27 First-Order Knowledge Compilation

Guy Van den Broeck (UCLA, US)

This talk gives an overview of recent developments in knowledge compilation for first-order logic. We focus the discussion on solving the (weighted) model counting task on first-order knowledge bases. This task received a lot of attention recently, as it underlies probabilistic database query evaluation [1], as well as inference and learning in statistical relational models [2]. The knowledge compilation approach faces several challenges: what is the appropriate first-order circuit language, like NNF, that supports tractable model counting, how to effectively compile first-order sentences into this language. We also discuss how

first-order compilation and counting are fundamentally different from their propositional counterparts.

References

1 Guy Van den Broeck and Dan Suciu. Query Processing on Probabilistic Data: A Survey, Foundations and Trends in Databases, Now Publishers, 2017.
2 Guy Van den Broeck. Lifted Inference and Learning in Statistical Relational Models, PhD thesis, KU Leuven, 2013

3.28 Inside d4

Jean-Marie Lagniez (IUT de Lens, FR)

Joint work of Jean-Marie Lagniez, Pierre Marquis

We present and evaluate a new compiler, called D4, targeting the Decision-DNNF language. As the state-of-the-art compilers C2D and Dsharp targeting the same language, D4 is a top-down tree-search algorithm exploring the space of propositional interpretations. D4 is based on the same ingredients as those considered in C2D and Dsharp (mainly, disjoint component analysis, conflict analysis and non-chronological backtracking, component caching). D4 also takes advantage of a dynamic decomposition approach based on hypergraph partitioning, used sparingly. Some simplification rules are also used to minimize the time spent in the partitioning steps and to promote the quality of the decompositions. Experiments show that the compilation times and the sizes of the Decision-DNNF representations computed by D4 are in many cases significantly lower than the ones obtained by C2D and Dsharp.

References

1 Jean-Marie Lagniez and Pierre Marquis, "An Improved Decision-DNNF Compiler," in *IJCAI 2017*, pp. 667-673, 2017.

4 System Presentations

4.1 The SDD and miniC2D Packages

Arthur Choi (UCLA, US) and Adnan Darwiche (UCLA, US)

URL http://reasoning.cs.ucla.edu/sdd/

The SDD Package is a system for constructing, manipulating and optimizing Sentential Decision Diagrams (SDDs). It further provides a compiler from CNF/DNF to SDDs, with dynamic variable and vtree search. The system further supports the apply operator (for conjoining/disjoining two SDDs), and support for queries such as (weighted) model counting, cardinality minimization, etc. In the upcoming version 2.0 release, the SDD Package will be open source, with improved dynamic search, and support for X-constrained vtrees. Moreover, the package will provide new, powerful primitives to facilitate the development of dynamic search algorithms.

The miniC2D package is a model counter, as well as a system for top-down compilation of CNFs/DNFs to (decision) SDDs. The package is open-source, and provides a framework for building new compilers and new model counters from SAT solvers.

The SDD Package is available at http://reasoning.cs.ucla.edu/sdd/ and the miniC2D package is available at http://reasoning.cs.ucla.edu/minic2d/.

4.2 Computing Problem Decompositions of Small Width with Local Improvement

Neha Lodha (TU Wien, AT)

Joint work of Johannes Fichte, Neha Lodha, Sebastian Ordyniak, Stefan Szeider

Many otherwise hard computational counting or optimisation tasks can be solved efficiently if a structural decomposition of small width of the instance is provided. Finding decompositions of optional or close to optimal width has been of interest for decades. In this talk we will show how we extended exact approaches as well as improved heuristically obtained upper bounds using local improvements. The two width parameters we focused on are branchwidth and treewidth. Our experiments show that the local improvement method for both branchwidth and treewidth are able to decrease the width of decompositions significantly.

References

1 Neha Lodha, Sebastian Ordyniak, Stefan Szeider. *A SAT Approach to Branchwidth*. SAT 2016
2 Neha Lodha, Sebastian Ordyniak, Stefan Szeider. *SAT-Encodings for Special Treewidth and Pathwidth*. SAT 2017
3 Johannes Klaus Fichte, Neha Lodha, Stefan Szeider. *SAT-Based Local Improvement for Finding Tree Decompositions of Small Width*. SAT 2017

4.3 Knowledge Compilation Tools

Joao Marques-Silva (University of Lisbon, PT)

Primer

Formula compilation by generation of prime implicates or implicants finds a wide range of applications in AI. Recent work on formula compilation by prime implicate/implicant generation often assumes a Conjunctive/Disjunctive Normal Form (CNF/DNF) representation. However, in many settings propositional formulae are naturally expressed in non-clausal form. Despite a large body of work on compilation of non-clausal formulae, in practice existing approaches can only be applied to fairly small formulae, containing at most a few hundred variables. We describe two novel approaches for the compilation of non-clausal formulae either with prime implicants or implicates, that is based on propositional Satisfiability (SAT) solving. These novel algorithms also find application when computing all prime implicates of

a CNF formula. The proposed approach is shown to allow the compilation of non-clausal formulae of size significantly larger than existing approaches.

HFLUBBER and IP-HORN

In knowledge compilation, the Horn Least Upper Bound (Horn LUB) of a formula F refers to the strongest Horn theory entailed by F. In [2], two SAT-based tools for computing Horn LUBs were proposed: HFLUBBER and IP-HORN. Both are based on successively querying a SAT solver in the lookout for so-called Horn prime-implicates.

References

1 Alessandro Previti, Alexey Ignatiev, Antonio Morgado and J. Marques-Silva. *Prime Compilation of Non-clausal Formulae*. IJCAI 2015
2 Carlos Mencía, Alessandro Previti and Joao Marques-Silva. *SAT-Based Horn Least Upper Bounds*. SAT 2015

4.4 Qute: A Dependency Learning QBF Solver

Friedrich Slivovsky (TU Wien, AT)

Joint work of Peitl, Tomáš; Slivovsky, Friedrich; Szeider, Stefan
Main reference Tomás Peitl, Friedrich Slivovsky, Stefan Szeider: "Dependency Learning for QBF", in Proc. of the Theory and Applications of Satisfiability Testing - SAT 2017 - 20th International Conference, Melbourne, VIC, Australia, August 28 - September 1, 2017, Proceedings, Lecture Notes in Computer Science, Vol. 10491, pp. 298–313, Springer, 2017.
URL http://dx.doi.org/10.1007/978-3-319-66263-3_19
URL https://www.ac.tuwien.ac.at/research/qute/

Qute is a solver for quantified Boolean formulas (QBFs) based on quantified conflict-driven constraint learning (QCDCL). Its key feature is a new technique for exploiting variable independence which we call *dependency learning* [1]. The resulting version of QCDCL enjoys improved propagation and increased flexibility in choosing variables for branching while retaining ordinary (long-distance) Q-resolution as its underlying proof system.

References

1 Tomáš Peitl, Friedrich Slivovsky, and Stefan Szeider. *Dependency Learning for QBF*. SAT 2017

4.5 ForcLift and PySDD

Wannes Meert (KU Leuven, BE)

ForcLift

- Method: Lifted /First-Order Weighted Model Counting
- Tasks:
  - Partition function of relational probabilistic model (e.g. MLN)
  - Marginal / conditional probabilities

```
Generate_GLB
Input: a set of clauses Σ = {C_1, C_2, ..., C_n}.
Output: a greatest Horn lower-bound of Σ.
begin
    L := the lexicographically first Horn–
         strengthening of Σ
    loop
        L' := lexicographically next Horn–
              strengthening of Σ
        if none exists then exit loop
        if L ⊨ L' then L := L'
    end loop
    remove subsumed clauses from L
    return L
end
```

Figure 1 Algorithm for generating a greates Horn lower-bound.

- Parameter and structure learning
- Available from https://github.com/UCLA-StarAI/Forclift

PySDD: SDD Python Wrapper

SDD wrapper formerly used in ProbLog (cython based). Seperate package is work in progress.

5 Open Problems

5.1 Revisiting Horn Approximations to Clausal Theories

Henry Kautz (University of Rochester, USA)

Horn upper-bounds and Horn lower-bounds

Let Σ be a set of clauses. The sets Σ_{lb} and Σ_{ub} of Horn clauses are respectively a Horn lower-bound and a Horn upper-bound of Σ iff

$$\mathcal{M}(\Sigma_{lb}) \subseteq \mathcal{M}(\Sigma) \subseteq \mathcal{M}(\Sigma_{ub}),$$

or, equivalently,

$$\Sigma_{lb} \models \Sigma \models \Sigma_{ub}.$$

Acylic Horn Theories

A set of Horn clauses is acyclic iff

- Its predicates can be numbered such that for every clause, the predicate in the positive literal (if any) has a higher number than any predicate in a negative literal.
- In each clause, every variable in a positive literal appears in a negative literal.

Finding Acyclic Horn Lower Bounds

Search through ways to transform the source theory to acyclic Horn by
- Deleting all but one positive literal in each clause.
- Deleting a minimal number of negative literals from the clauses such that the resulting theory is non-recursive.
- Return any LB found this way that does not strictly entail some other LB.

Question 1: is the result always a greates lower bound?

Compiling Prolog to ASP

Prolog extends Horn logic with negation as failure.
- Negation as failure operator may appear in front of literals in the body of a clause.
- Undecidable
- Some programs cannot be given coherent semantics.

Answer set programming (ASP) restricts Prolog to stratified theories.
- Decidable
- Each program has stable set semantics.

Question 2: Can ASP GLBs of Prolog programs be found by a version of Generate_GLB?

5.2 Bounds on the DNNF size of Disjoint Sums

Stefan Mengel (CNRS, CRIL – Lens FR)

Consider Boolean functions $f(x_1, \ldots, x_k)$ and $f(y_1, \ldots, y_r)$ for which we have DNNF lower bounds, where $\{x_1, \ldots, x_k\} \cap \{y_1, \ldots, y_r\} = \emptyset$. What can we say about the DNNF size of $f(x_1, \ldots, x_k) \wedge f(y_1, \ldots, y_r)$?

5.3 Compilation of CNFs into Decision DNNFs parameterized by Incidence Treewidth

Igor Razgon (Birkbeck, University of London, GB)

The incidence graph of a CNF F is the undirected bipartite graph which has the variables and clauses of F for vertices, and a variable v is adjacent to a clause C if v occurs in C. It is known that CNFs admit FPT-size compilation into d-DNNFs, parameterized by incidence treewidth. However, it is open whether *CNFs admit FPT-size compilation into decision DNNFs, parameterized by incidence treewidth.*

A closely related question is the following: *Do unsatisfiable CNFs admit FPT-size resolution refutations, parameterized by incidence treewidth?*

5.4 Smoothing in Subquadratic Time

Guy Van den Broeck (UCLA, US)

An NNF is *smooth* if, for every disjunction C in the NNF, each disjunct of C mentions the same variables. That is, if $C_1, \ldots, C_n$ are the children of or-node C, then $vars(C_i) = vars(C_j)$, where $1 \leq i, j \leq n$.

A (d-)DNNF can be transformed in into a smooth (d-)DNNF in quadratic time. *But can smoothing be performed in subquadratic time?*

5.5 On the VC-dimension of DNNF and SDD

Frederic Koriche (Artois University – Lens, FR)

A central problem in learning theory is to determine the sample complexity of concept classes. Roughly, the sample complexity of a concept class C is the number of examples required to output a concept (from the given class) with good generalization behavior. This complexity measure is intrinsically related to the VC-dimension of C, which is the maximum size of any set of examples that can be labeled in any possible way using concepts taken from C. Although the VC-dimension of binary decision trees and DNF formulas is well-known, an interesting open question is to identify (or at least, to approximate) the sample complexity of nontrivial subclasses of DNNF and SDD (e.g. decision DNNF formulas of bounded depth, SDD of size polynomial in the dimension of examples, etc.)

5.6 First Order Model Counting

Dan Suciu (University of Washington – Seattle, US)

$\#P_1$ is the class of functions in $\#P$ over a unary input alphabet (also called *tally problems*).

We know that there exist (symmetric) First Order Model Counting (FOMC) queries that $\#P_1$-hard. Open problems:

- Give an example of query where FOMC is $\#P_1$-hard.
- Characterize the class of FOMC queries in P or the class of queries that is $\#P_1$-complete.

Participants

- Antoine Amarilli
Télécom ParisTech – Paris, FR
- Lameck Mbangula Amugongo
Namibia Univ. of Science & Technology – Windhoek, NA
- Paul Beame
University of Washington – Seattle, US
- Arpita Biswas
Indian Institute of Science – Bangalore, IN
- Pierre Bourhis
CNRS – Lille, FR
- Simone Bova
TU Wien – Vienna, AT
- Florent Capelli
INRIA – Lille, FR
- Ondrej Cepek
Charles University – Prague, CZ
- Zaineb Chelly Dagdia
Aberystwyth University – Aberystwyth, GB
- Arthur Choi
UCLA – Los Angeles, US
- YooJung Choi
UCLA – Los Angeles, US
- Adnan Darwiche
UCLA – Los Angeles, US
- Ronald de Haan
University of Amsterdam – Amsterdam, NL
- Hélène Fargier
University of Toulouse – Toulouse, FR
- Robert Ganian
TU Wien – Vienna, AT
- Martin Grohe
RWTH Aachen – Aachen, DE
- Henry A. Kautz
University of Rochester – Rochester, US
- Batya Kenig
Technion – Haifa, IL
- Frederic Koriche
Artois University – Lens, FR
- Oliver Kullmann
Swansea University – Swansea, GB
- Jean-Marie Lagniez
Artois University – Lens, FR
- Neha Lodha
TU Wien – Vienna, AT
- Meena Mahajan
Institute of Mathematical Sciences – Chennai, IN
- Joao Marques-Silva
University of Lisbon – Lisbon, PT
- Pierre Marquis
Artois University – Lens, FR
- Wannes Meert
KU Leuven – Leuven, BE
- Stefan Mengel
CNRS, CRIL – Lens FR
- Shin-ichi Minato
Hokkaido University – Hokkaido, JP
- Alexandre Niveau
Caen University – Caen, FR
- Jakob Nordström
KTH Royal Institute of Technology – Stockholm, SE
- Dan Olteanu
University of Oxford – Oxford, GB
- Igor Razgon
Birkbeck, University of London – London, GB
- Subhrajit Roy
IBM Research – Melbourne, AU
- Lisset Y. Salinas Pinacho
ifib – Bremen, DE
- Scott Sanner
University of Toronto – Toronto, CA
- Rahul Santhanam
University of Oxford – Oxford, GB
- Marco Schaerf
Sapienza University of Rome – Rome, IT
- Pierre Senellart
ENS – Paris, FR
- Laurent Simon
University of Bordeaux – Bordeaux, FR
- Friedrich Slivovsky
TU Wien – Vienna, AT
- Dan Suciu
University of Washington – Seattle, US
- Stefan Szeider
TU Wien – Vienna, AT
- Guy Van den Broeck
UCLA – Los Angeles, US